

AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A method, comprising:

requesting access to a shared resource for a first process having a first local priority, the first process having a corresponding first semaphore;

determining if a second process is simultaneously requesting access to the shared whether the resource is being accessed by a second process, the second process having a second local priority corresponding second semaphore;

and

denying the first process access to the resource if the resource is being accessed by the second process as indicated by a lock on the resource, wherein the lock is indicated at the second semaphore if the second process is simultaneously requesting access to the shared resource, then granting access one of the first priority and the second priority having a higher local priority.

2. (Currently Amended) The method of claim 1, wherein the local priority is fixed for each of the first and the second process further comprising the first process having a corresponding first local priority and the second process having a corresponding second local priority.

3. (Currently Amended) The method of claim 1, ~~additionally further comprising if the second process is not simultaneously requesting access to the shared resource, then:~~

~~determining if the second process currently has a lock on the shared resource;~~
~~if the second process currently has a lock on the shared resource, then denying the first process access to the shared resource; and~~
~~if the second process does not have a lock on the shared resource, then granting the first process access to the shared resource if the resource is not being accessed by the second process as indicated at the second semaphore.~~

Claims 4-23 (Cancelled)

24. (Currently Amended) A machine-readable medium having stored thereon data representing sequences ~~sets~~ of instructions, ~~the sequences of instructions which, when executed by a processor machine, cause the processor machine to perform the following:~~

~~request access to a shared resource for a first process having a first local priority, the first process having a corresponding first semaphore;~~
~~determine if a second process is simultaneously requesting access to the shared whether the resource is being accessed by a second process, the second process having a second local priority corresponding first semaphore; and~~
~~deny the first process access to the resource if the resource is being accessed by the second process a indicated by a lock on the resource, wherein the lock~~

is indicated at the second semaphore if the second process simultaneously requests access to the shared resource, then grant access one of the first priority and the second priority having a higher local priority.

25. (Currently Amended) The machine-readable medium of claim 24, wherein the local priority is fixed for each of the first and the second process further comprising the first process having a corresponding first local priority and the second process having a corresponding second local priority.
26. (Currently Amended) The machine-readable medium of claim 24, additionally comprising if the second process is not simultaneously requesting access to the shared resource, then additionally comprising sequences of instructions which, when executed by a processor, cause the processor to perform wherein the sets of instruction which, when executed by the machine, further cause the machine to: determine if the second process currently has a lock on the shared resource; if the second process currently has a lock on the shared resource, then deny the first process access to the shared resource; and if the second process does not have a lock on the shared resource, then grant the first process access to the shared resource if the resource is not being accessed by the second process as indicated at the second semaphore.
27. (Currently Amended) An apparatus, comprising:
a central processing unit having a processor to execute a plurality of processes including a first process and a second process; and

the processor is further to at least one processor; and
~~a machine readable medium having instructions encoded thereon, which when~~
~~executed by the processor, are capable of directing the processor to:~~

~~request access to a shared resource for a~~the first process having a first local priority, the first process having a corresponding first semaphore;

~~determine if a second process is simultaneously requesting access to the shared~~whether the resource is being accessed by the second process, the second process having a second local priority corresponding second semaphore; and

deny the first process access to the resource if the resource is being accessed by the second process as indicated by a lock on the resource, wherein the lock is indicated at the second semaphore; if the second process simultaneously requests access to the shared resource, then grant access one of the first priority and the second priority having a higher local priority.

28. (Currently Amended) The apparatus of claim 27, wherein the local priority is fixed for each of the first and the second process further comprising the first process having a corresponding first local priority and the second process having a corresponding second local priority.
29. (Currently Amended) The apparatus of claim 27, wherein the processor is further to additionally comprising if the second process is not simultaneously requesting

~~access to the shared resource, then additionally encoded instructions which, when executed by a processor, are capable of causing the processor to:~~

~~determine if the second process currently has a lock on the shared resource;~~

~~if the second process currently has a lock on the shared resource, then deny the first process access to the shared resource; and~~

~~if the second process does not have a lock on the shared resource, then grant the first process access to the shared resource if the resource is not being accessed by the second process as indicated at the second semaphore.~~

Claims 30-32 (Cancelled)

33. (New) The method of claim 2, further comprising:

determining if access to the resource is simultaneously being requested by the second process; and

granting access to the resource to one of the first process and the second process having a higher local priority of the first local priority and the second local priority.

34. (New) The machine-readable medium of claim 25, wherein the sets of instructions which, when executed by the machine, further cause the machine to:

determine if access to the resource is simultaneously being requested by the second process; and

grant access to the resource to one of the first process and the second process
having a higher local priority of the first local priority and the second local
priority.

35. (New) The apparatus of claim 28, wherein the processor is further to:

determine if access to the resource is simultaneously being requested by the
second process; and

grant access to the resource to one of the first process and the second process
having a higher local priority of the first local priority and the second local
priority.
36. (New) A system, comprising:

a memory having a plurality of resources being accessed by a plurality of
processes, and a plurality of semaphores associated with the plurality of
processes; and

a processor coupled with the memory, wherein the processor is capable of
executing the plurality of processes, the processor is further to

request access to a resource for a first process of the plurality of processes,
the first process having a corresponding first semaphore of the
plurality of semaphores;

determine whether the resource is being accessed by a second process of
the plurality of processes, the second process having a
corresponding second semaphore of the plurality of semaphores;

and

deny the first process access to the resource if the resource is being accessed by the second process as indicated by a lock on the resource, wherein the lock is indicated at the second semaphore.

37. (New) The system of claim 36, further comprising the first process having a corresponding first local priority and the second process having a corresponding second local priority.
38. (New) The system of claim 36, wherein the processor is further to grant the first process access to the resource if the resource is not being accessed by the second process as indicated at the second semaphore.
39. (New) The system of claim 37, wherein the processor is further to:
determine if access to the resource is simultaneously being requested by the second process; and
grant access to the resource to one of the first process and the second process having a higher local priority of the first local priority and the second local priority.